System-Directed Resilience for Exascale Platforms

Proposal No. FY-10-1017

1 Overview of the Problem and Idea

Resilience on MPP systems has traditionally been the responsibility of the application, with the primary tool being application-directed checkpoints. However, as systems continue to increase in size and complexity, the viability of application-directed checkpoint as a solution decreases. Recent studies performed at SNL projected that as systems grow beyond 100,000 components, a combination of factors lead to checkpoint overheads in excess of 50%. In this project, we will investigate critical changes required in MPP systems software to support system-directed resilience. The goal is to provide efficient, application-transparent resilience through coordinated use of system resources. The primary research topics focus around the problem of continuous computing in the event of a component failure. A preliminary list of required new capabilities include:

- **Application Quiescence:** the ability to suspend CPU, network, and storage services used by an individual application without interfering with the progress of other applications;
- **State Management:** the ability to identify, extract, and manage application state in a transparent, efficient, and non-intrusive way; and
- **Fault Recovery:** the ability to transparently replace a failed component without restarting the entire application.

2 R&D Accomplishments: Progress Toward Milestones

The primary goals of FY'09 were to understand application requirements for resilience and begin to explore the viability of alternative resilience approaches on capability-class systems. Toward those goals, we completed an extensive set of memory-characterizations studies, we developed tools to identify periods in an applications execution where quiescence has minimal impact, and we began to explore a hybrid approach to resilience that combines traditional checkpointing with redundant computation.

The memory-characterization study satisfied our FY'09 milestones to characterize application behavior and identify application critical state. Our software identifies all memory blocks that changed within a given time period. The goal is to test the viability of incremental-checkpoint approaches for important ASC applications. Our characterizations include HPCCG, CTH, Sage, and LAMPS. In the second half of FY'09, we will publish these results in a SAND report.

To satisfy our "investigate quiescence options" milestone, we developed code to collect network-related traces and we are in the process of collecting I/O traces. The goal is to correlate data for network and I/O activity to identify "windows of opportunity" where quiescence has little or no impact on external services. We are also actively involved in discussions with Cray about their solution for quiescence of the XT series systems. Cray is particularly interested in how to suspend applications that have active connections to file systems and other shared services. Our work on correlating network and I/O data could help checkpoint-based solutions make better decisions about when to perform a checkpoint.

We also completed development of an MPI-based *partial-redundant* computation-scheme that duplicates some processes in an application to reduce the probability of failure for the entire application. This approach, when combined with traditional checkpoint-based resilience, reduces the frequency of checkpoints allowing checkpoint-based approaches to scale to much larger application sizes. This approach also provides some user-control over the "reliability" of the application by controlling the level of redundancy. One goal for FY'10 is to develop appropriate metrics for

comparison between the overheads associated with a partial-redundant approach when compared to traditional checkpoints. This accomplishment was not specifically called out in our original proposal, but we believe this approach shows promise as a viable alternative to pure checkpoint-based methods.

3 Proposed R&D

In FY'09, we identified three approaches that merit further investigation on capability-class systems: incremental checkpointing, diskless checkpointing, and partial-redundant computation. Incremental checkpointing and diskless checkpointing have a long history of research [5, 1], but have not successfully been applied to HPC systems largely due to overheads associated with state management and quiescence. Partial-redundant computation is a new approach that requires new capabilities for node replacement and an understanding of the overheads added by additional resources. To explore the viability of these candidate solutions, we must perform extensive research in quiescence, state management, and recovery.

To efficiently quiesce a large-scale application not only requires cooperation among the application processes, it also requires integration and cooperation with shared services like the network, scheduler, and storage system. Our ongoing work to correlate network and I/O traces will help identify periods during an application's execution where quiescence will have minimal impact on the overall system. We will continue to explore ways to reduce the impact of quiescence when it is needed, as well as exploring alternative resilience approaches that do not require quiescence.

Efficient state management is perhaps the largest performance challenge for exascale resilience. On todays systems, the I/O associated with checkpoint data (for application-directed checkpoints) accounts for nearly 80% of the total I/O of the system [4]. Since I/O is such a large bottleneck for traditional resilience approaches, we will focus on approaches that significantly reduce the I/O required to maintain a resilient application. One product of our memory characterization work was a low-overhead hash-based tool for identifying all memory blocks that changed since the last checkpoint. If incremental checkpoints are viable, this tool will need to become part of the resilience software. We also plan to explore compression, extensive network caching [2, 3], and diskless approaches as ways to reduce the I/O burden of resilience on the system.

The third research area is system support for recovery. Applications currently use a "roll-back" approach that requires the user to kill the application after a single node failure, re-allocate all the resources for the application, load the data from the last checkpoint, and finally continue computing. This approach is time-consuming, wastes valuable resources, and is unfair to applications that have to re-submit their failed job through a batch queue system. Our solution to use partial-redundancy, described in Section 2, addresses this at the MPI layer. We developed a method that allows two MPI processes to behave as one. Both processes receive messages from other nodes, but only the "active" process sends messages. They both do computation so their state should be identical. When the active process fails, the redundant node takes the place of the failed node and computation continues. There are two advantages to this approach: there is no roll-back penalty for failure, and the overhead of replacing a failed node is minimal. The disadvantages are that it requires compute resources and memory that could be applied to the problem and that there is additional network traffic generated for the redundant nodes. There are also unresolved issues with how to deal with nodes that require other shared resources (e.g., a file system). In FY'10, we will continue to explore this option and develop metrics to help understand the viability of this approach for large-scale applications.

3.1 Key Project Goals and Milestones:

Earlier work showed that traditional checkpoint approaches are not practical for the scale of applications expected on the next generation of capability-class systems [3]. The technical goals of this project are to identify and assess the viability of alternative approaches to resilience for extreme-scale applications, and perform the necessary R&D to prototype and evaluate these candidate solutions.

Goal	Milestone	Completion Date
Preliminary Work		
	Investigate options for low-overhead application quiescence	02/01/2009 (ongoing)
	Investigate system software to support dynamic node allocation, network/os virtualization, and MPI node recovery.	02/01/2009 (ongoing)
	Investigate options for diskless checkpoint/recovery	06/01/2009
	Investigate/define RAS system requirements (APIs) to support resilience	10/01/2009
Viability of Incremental Checkpoints		
	Develop code to automatically identify critical state	09/01/2009 (Complete)
	Complete memory characterization of selected applications	03/01/2009 (Complete)
	Publish report on application memory characterizations	04/01/2009 (Delayed)
	Identify methods for incremental checkpoint recovery	06/01/2009
	Complete prototype of incremental checkpoint recovery	11/01/2009
	Publish report on performance evaluation of incremental approach	02/01/2009
Viability of Diskless Approaches		
	Develop metrics for comparison of diskless methods to traditional chkpt	10/01/2009
	Evaluate performance of Pongo: Cray library for diskless chkpt/recovery	12/01/2009
	Publish report (with Cray) on Pongo performance	02/01/2010
Viability of Partial-Redundant Computation		
	Develop metrics for comparison of partial-redundant approach to traditional chkpt	10/01/2009
	Develop MPI-based prototype for redundant computation	05/01/2009 (Complete)
	Evaluate performance of partial redundant computation on representative apps	11/01/2009
	Publish report on viability of partial-redundant computation	02/01/2010
Integrated Solution to Resilience		
	Publish paper on quantitative comparison of different resilience approaches	04/01/2010
	Complete design of system that supports RAS integration, low-overhead quiescence, and a provides "best-of" solutions for resilience	10/01/2010
	Complete prototype of integrated resilience software	04/01/2011
	Publish report on performance evaluation of integrated software	07/01/2011
Project Completion		
	Publish final report	10/01/2011
	Publish final results and conclusions in journal	12/01/2011

3.2 Leading Edge Nature of Work:

This project is about developing resilience solutions that, while applicable to modern systems, are designed with the next-generation in mind. Scalability is paramount! In addition, we are addressing some solutions that require substantial changes to the system software and perhaps to the underlying operating system. Our expertise and tight-collaboration with ongoing lightweight-kernel research puts us in a unique position to understand the impact of these changes, and it puts on the fast-track for getting these ideas accepted by HPC vendors.

3.3 Technical Risk and Likelihood of Success:

The challenge (and inherent risk) in a systems-driven approach to resilience for MPP systems is (and always has been) performance. A viable solution needs to be sensitive to resource requirements, scale, and performance issues that are particularly demanding in exascale systems. Although there is risk that a system-directed approach may still impose significant overhead on the application, we are confident that our combined expertise on MPP operating systems, networking, and storage systems will lead to a solution that is efficient and reduces the burden of resilience on the application developer.

3.4 Unusual Budget Items (if any)

A large part of this work involves R&D at the system-software level for leading-edge systems. Sandia has small-scale experimental systems that are often used for these purposes. A portion of our expenses may go toward offsetting the overhead costs (e.g., power, system administration, etc.) associated with maintaining these systems. However, we do not expect these expenses to exceed 25% of the total planned budget.

4 Strategic Alignment and Potential Benefit

4.1 Relevance to Missions:

This project has direct relevance to the Science, Technology, and Engineering mission of Sandia National Laboratories, if successful, it will have a direct impact on applications in virtually all areas of advanced computing, particularly efforts in the Enabling Predictive Simulation investment area.

Our approach represents a fundamental change in the way MPP systems support resilience. If successful, our approach will significantly reduce the burden of resilience on the application developer, simplifying the development process for the scientist. This effort could also have a tremendous impact on performance and resource utilization by reducing (by orders of magnitude) the network and storage requirements for fault tolerance.

4.2 Communication of Results:

We will publish the initial studies and design documents for all aspects of this project in SAND reports. As appropriate, we will also present some of the studies at ACM/IEEE conferences as position papers or at user-group meetings (e.g., CUG) as works in progress. We will target well-known ACM/IEEE supercomputing conferences (e.g., IPDPS, SC, Cluster) for publication of performance analysis and prototype results. As we approach completion of the project, we will publish detailed results, design, and lessons learned in well known Journals such as the Journal of Parallel and Distributed Computing (JPDC) and IEEE Transactions on Parallel and Distributed Systems (TPDS).

In addition to the standard publications such as Conferences, Journals, and user-group meetings; we will actively pursue vendor participation to incorporate our research into a commercialized product. The ultimate goal is to see the results of our work continue beyond the life of the project. Our best chance at that is through vendor participation and adoption.

References

[1] Christian Engelmann and Al Geist. A diskless checkpointing algorithm for super-scale architectures applied to the fast fourier transform. In *In Proceedings of the International Workshop on Challenges of Large Applications in Distributed Environments*, pages 47–52, Seattle, WA, June 2003. IEEE Computer Society Press.

[2] Ron A. Oldfield. Investigating lightweight storage and overlay networks for fault tolerence. In *Proceedings of the High Availability and Performance Computing Workshop*, Santa Fe, NM, October 2006.

[3] Ron A. Oldfield, Sarala Arunagiri, Patricia J. Teller, Seetharami Seelam, Rolf Riesen, Maria Ruiz Varela, and Philip C. Roth. Modeling the impact of checkpoints on next-generation systems. In *Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies*, San Diego, CA, September 2007.

[4] Fabrizio Petrini and Kei Davis. Tutorial: Achieving Usability and Efficiency in Large-Scale Parallel Computing Systems, August 31, 2004. Euro-Par 2004, Pisa, Italy.

[5] James S. Plank, Youngbae Kim, and Jack J. Dongarra. Fault-tolerant matrix operations for networks of workstations using diskless checkpointing. *Journal of Parallel and Distributed Computing*, 43(2):125–138, June 1997.